Vision-based grasping of unknown objects to improve disabled people autonomy.

C. Dune, A. Remazeilles, E. Marchand, Cédric Leroux

Vision-based grasping of unknown objects to improve disabled persons autonomy

Claire Dune*†, Anthony Remazeilles*, Eric Marchand† and Christophe Leroux*
*CEA, LIST, 18 route du panorama, BP6, Fontenay aux Roses F-92265, France
Email: firstname.lastname@cea.fr
†INRIA, IRISA, Lagadic Project, F-35000 Rennes, France,
Email: firstname.lastname@irisa.fr

***Abstract*— This paper presents our contribution to vision based robotic assistance for people with disabilities. The rehabilitative robotic arms currently available on the market are directly controlled by adaptive devices, which lead to increasing strain on the user's disability. To reduce the need for user's actions, we propose here several vision-based solutions to automatize the grasping of unknown objects. Neither appearance data bases nor object models are considered. All the needed information is computed on line. This paper focuses on the positioning of the camera and the gripper approach. For each of those two steps, two alternative solutions are provided. All the methods have been tested and validated on robotics cells. Some have already been integrated into our mobile robot SAM.**

I. INTRODUCTION

This work relates to robotic assistance for disable people, where autonomous robotic systems are designed to compensate for a human motor disability. We propose solutions for the grasping of any object within a domestic environment such as an apartment. Providing a robust, generic and easy-to-use solution to improve the user's interaction with their personal environment would largely increase their autonomy.

Contrary to an industrial environment [19], the domestic environment is highly unstructured. Thus, the robotic system needs exterioceptive sensors to adapt its behavior to the current situation. Vision sensors are almost always used: this sensor is quite cheap, the acquired information is very rich, and it can even be directly used as feedback for the user.

A. State of the art

Before starting a grasping procedure, a robotic system first needs to extract information on the object from the visual input. In order to handle any object shape and appearance, it is necessary to make some assumptions on the situations the robot can handle.

Some approaches propose to constrain the possible locations for the object. For example, [11] assumes that the scene is known and uses a simple image difference with the known background to localize the object. The project FRIEND II reduces the grasping area to a tactile tray fixed on an instrumented wheelchair [24].

Since the user would like to operate anywhere in his home, it is difficult to constrain the grasping place ; assumptions must then be made on the objects themselves. Some solutions rely on a data base of objects which is used to recognize the scene observed by the camera. In [14], the object recognition and pose estimation is performed by comparing SIFT descriptors [17] and color histograms with the database. Object tracking methods like [15, 5] suppose that an object model is known (respectively a sparse $3D$ model and a structured one).

Instead of requiring the knowledge of all possible objects several methods propose to infer the object characteristics or shape in order to get a set of object categories that are then used to guide the robot toward the grasping position. In [20], a set of rendered $3D$ models are used as a training database. A supervised learning stage enables an object to be associated with one of the five obtained categories, and from there selects the best grasping position. The MOVAID project [22] uses a mixed fuzzy logic/neural network module to select the best grasping position.

Naturally, the user expects to be able to grasp any object in his environment. Nevertheless, no machine learning or object recognition technique can succeed in handling every kind of object. It is thus necessary to provide solutions to deal with unknown objects, at least as a complement to these methods. In this context, several approaches propose to infer the object characteristics from its observed shape. The $2D$ structure of the object can be used to determine the grasping position, such as its skeleton [11] or its $2D$ moments [19]. Some approaches rely on implicit $3D$ functions to model the object's $3D$ shape, using active vision to refine the estimated parameters [25, 10].

In most of the robotic systems, the camera is embedded onto the arm gripper (*eye-in-hand* configuration) and the object is supposed to be directly within the camera field of view (FOV). Nevertheless, the perception of the environment around the arm is strongly restricted, and there is little chance that the above requirement is met, especially when the arm is mounted on a mobile unit. Few methods address this problem. It is usually solved by using an external additional camera (*eye-to-hand* configuration). In [12] an initialization step ensures that a moving object detected by the eye-to-hand camera falls within the embedded camera's FOV. [14] adds a wide FOV stereo rig to orientate an eye-in-hand stereo rig toward the object direction.

B. Our system philosophy

Our robotic system has been designed to observe the following constraints: (i) no assumption is made on the scene

structure surrounding the object to grasp, (ii) no *a priori* information on the object appearance (no $3D$ model, no image database) is used (iii) the user's actions are reduced to a minimum.

In this paper, we propose two alternate solutions to address situations where the object is not directly inside the embedded camera FOV (section II). We then investigate the automatic positioning of the arm in front of the object (section III).

Since there is not a unique solution to perform vision-based grasping, it is possible to provide several concurrent methods, with different physical architectures and algorithmic assumptions. The best solution can then be selected depending on the user's situation, and his personal preferences.

The current design of our robot SAM [18] is a result of discussions with end users, especially from the APPROCHE[1] group. One of their main concerns was to avoid creating a bulky wheelchair: some users were indeed complaining about the increased size of a wheelchair with an embedded arm, preventing them from moving freely in their apartment [8].

SAM (see Fig. 1) is made of a mobile platform (MPM470[2]) and a MANUS arm[3]. The mobile unit offers ready-to-use solutions for self-localization and navigation (thus we suppose in this paper that the desired object is reachable by the arm). The MANUS arm is the most widespread arm within the rehabilitation field [1]. The user interacts with the robot through a remote HMI designed to minimize the user's action.

II. Arm orientation toward the object direction

The very first step to start any vision-based grasping is to get the object within the embedded camera FOV. We propose to use an eye-to-hand camera to get a global view of the environment. A single click on this view gives SAM enough information to move its eye-in-hand camera so that it's FOV holds the object. Two alternative solutions are described, using respectively a catadioptric sensor and a pinhole camera.

A. Arm positioning with a catadioptric sensor

The appeal of an omnidirectional camera is that a single acquisition gives a 360° view of the environment. The mirror in our sensor has been worked out to get a vertical FOV wide enough to see an object from the floor up to 1.30 m high [4].

The omnidirectional camera is mounted on the MANUS shoulder (see Fig. 2(a)), *i.e.* its first joint. The direction of the first axis remains constant within the panoramic view. Furthermore, there is a direct mapping between an image x-coordinate x_{q_p} and the corresponding first joint angle q_p. Let x_{q_0} be the constant first joint projection onto the panoramic view. Then the motion to perform, such that this joint points toward the selected direction, is:

$$\Delta_q = \frac{2\pi}{x_M}\left(x_{q_p} - x_{q_0}\right), \tag{1}$$

where x_M denotes the horizontal length of the panoramic view.

[1]association promoting the use of robotics platform by disabled people
[2]designed by Neobotix: http://www.neobotix.de
[3]designed bu ExactDynamics: http://www.exactdynamics.nl/

Fig. 1. SAM: a Manus arm mounted onto the MPM470 mobile platform.

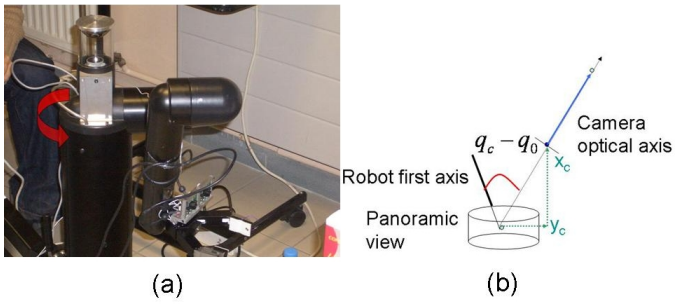

Fig. 2. Panoramic-based arm positioning: (a) the omnidirectional camera mounted onto the MANUS first axis, (b) the panoramic-embedded camera's relation when the second one is correctly aligned.

The eye-in-hand camera's optical axis is to be aligned with the axis passing the optical centers of the two cameras, so that the embedded camera acts as if it was rigidly linked to a virtual axis centered on the base frame. As soon as this alignment is achieved, the motion to perform to see the direction indicated by the user with the embedded camera is:

$$\begin{aligned} q_0^* &= q_0 + \Delta_q - (q_c - q_0) \\ &= 2q_0 + \tfrac{2\pi}{x_M}\Delta_x - \arctan\left(\tfrac{x_c}{y_c}\right), \end{aligned} \tag{2}$$

where (x_c, y_c) are the embedded camera frame center coordinates expressed in the base frame. This method ensures that the vertical $3D$ line going through the indicated point is centered in the eye-in-hand view.

Figures 3 and 4 illustrate this method. The left image of Fig. 3 is the initial embedded camera FOV. The desired object (a coffee box) is not visible. Fig. 4 is the panoramic view provided to the user. The right picture of Fig. 3 is the view given by the camera after the positioning of the arm onto the object.

This method has been assessed and verified during one month within four French medical centers[4], by 24 valid and 20 tetraplegic people. Even though the user feedback was globally positive, some constraints were considered as drawbacks by some people. The first complaint was that the image resolution

[4]CRF Coubert, CHU Reims, Center Calvé at Berck sur Mer, and CHU Raymond Poincaré at Garches

Fig. 3. Panoramic-based arm positioning: images acquired by the embedded camera before and after the motion.

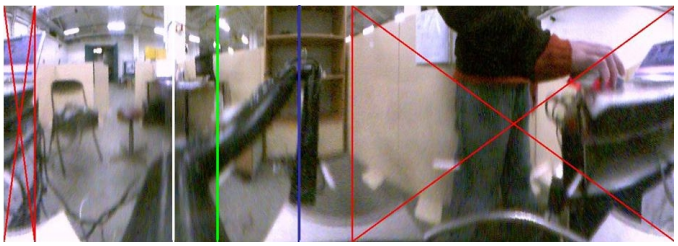

Fig. 4. Panoramic view provided to the user before the arm positioning. Red crosses: positions that can not reach the first joint. White line: first axis position. Blue line: initial FOV of the camera (left picture of Fig 3). Green line: desired camera direction, given by the user with one click. After the arm positioning, the embedded camera gives the right picture of Fig 3.

is not sharp enough, especially on the lower part of the image-corresponding to the central area of the acquired view, described by fewer pixels. Another complaint was that this solution does not control the gripper's height, and may need additional user action to adjust the gripper vertically to see the object.

B. Arm positioning with an eye-to-hand pinhole camera

In this section, the eye-to-hand imaging is done by a pinhole camera. Given the user's click on this view and the calibration of the system, the object coordinates along the $\vec{x}$ and $\vec{y}$ axes within the eye-to-hand camera frame are directly obtained. However, the depth of the object remains unknown, and thus we get a set of candidate positions within the eye-in-hand camera frame corresponding to an epipolar line. The method proposed here consists of scanning this line with the eye-in-hand camera and detecting the location of the object by image processing [7].

1) Surfing on the epipole: The geometrical relations describing a scene observed by two cameras can be summarized by the essential matrix ${}^2\mathbf{E}_1$:

$$ {}^2\mathbf{p}^\top {}^2\mathbf{E}_1 {}^1\mathbf{p} = 0, \tag{3} $$

which indicates that the point corresponding to the clicked point ${}^1\mathbf{p}$ belongs to a line in the eye-in-hand view, the *epipolar line* ${}^2\mathbf{E}_1 {}^1\mathbf{p}$. The essential matrix is directly defined by the relative position of the two cameras. Thus, if the defined line is scanned by the second camera, the corresponding point ${}^2\mathbf{p}$ will necessarily be observed.

The epipolar line is scanned using visual servoing. Visual servoing aims to reduce the difference $\mathbf{e_s} = \mathbf{s} - \mathbf{s}^*$ between a visual feature value $\mathbf{s}$ observed by a camera, and its desired

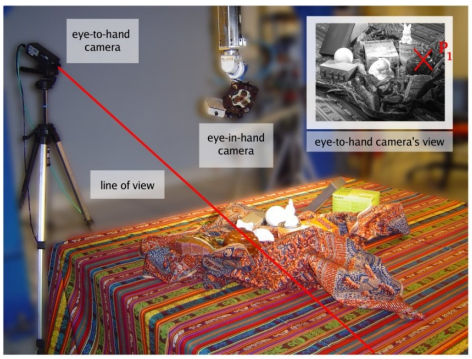

Fig. 5. Experimental setup, with a cluttered scene. The red line, defined by the user click, is the epipolar line that is covered by the embedded camera.

value $\mathbf{s}^*$. This minimization is performed by moving the camera with a velocity deduced from [3]:

$$ \tau_c = -\lambda \mathbf{L_s}^+ \mathbf{e_s}, \tag{4} $$

where λ is a positive scalar, and $\mathbf{L_s}$ is the interaction matrix linking the variation of the feature position to the motion of the camera. In order to scan the line, we use a redundant control law involving two tasks. The first task, $\mathbf{e}_1$, controls the orientation of the camera (*i.e.* the arm) so that the epipolar line stays horizontal and centered in the embedded view, while the second task, $\mathbf{e}_2$, handles the camera motion along this line. The control law is [3]:

$$ \tau_c = -\lambda_1 \widehat{\mathbf{L_1}}^+ \mathbf{e_1} - \lambda_2 \mathbf{P} \widehat{\mathbf{L_2}}^+ \mathbf{e_2} \tag{5} $$

The redundancy framework ensures that the epipolar line centering (primary task) remains satisfied by requiring the line covering task to operate onto the null space of $\mathbf{L_1}$.

2) Bayesian object detection: The visual appearance of the object is defined by the region around the user's click in the eye-to-hand view. The object's location is thus obtained by comparing this description with the ones acquired by the eye-in-hand camera during the line scanning.

The reference and all the candidate zones are characterized by SIFT descriptors [17], and each couple reference-candidate view is searched for matches. Each match gives a confidence in the underlying object depth. Finally the depth having the highest score is associated with the object. The object is finally brought back to the eye-in-hand camera FOV.

3) Experimental results: This method has been applied experimentally and validated on a robotic cell (the experimental setup is displayed on fig. 5). Figure 6 presents views before, during and after the arm navigation. The object is always correctly brought inside the camera FOV.

III. Grasping unknown objects

Both of the previous stages ensure that the camera positioning requirement (defined at the end of Sec. I-A) is met. This section proposes two different solutions for the autonomous object grasping. The first one (section III-A), based on a stereo virtual visual servoing, can handle textured objects. The grasping strategy consists in servoing the translational degrees

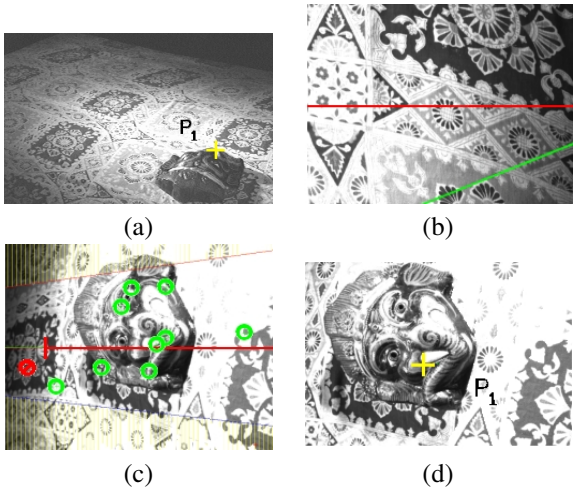

Fig. 6. Method illustration : (a) eye-to-hand view, with the user click (b) initial eye-in-hand view, with the current epipolar line in green and its desired position in red, (c) view during line scanning, (d) final FOV of the camera.

of the arm to bring the gripper in front of the object. The second one, based on an active estimation of the object shape (section III-B) leads to a more accurate grasping position, but needs an additional exploration step.

A. Stereovision-based object grasping

This first solution compensates for the lack of information on the object to grasp by embedding a stereo rig on the gripper (see Fig. 7). It relies on a tracking method estimating at each iteration the object pose within the camera frame, in order to guide the arm just in front of the object. This pose estimation uses the virtual visual servoing framework that reuses the principle of visual servoing (see eq. (4)). The description made in [5] uses contours as visual information ; in our case, we consider Harris points.

1) Sparse Object Model estimation: The virtual visual servoing needs an object model to realize the estimation of the object pose ; information that we do not have. Thus, an estimation of this model has to be performed on-line. The advantage of a stereo rig is that $3D$ information can be directly extracted without moving the arm.

The input of the process is a box surrounding the object defined by the user on a remote display-which can be done in only two image clicks. First, Harris points are extracted from the region of interest, and their relatives are searched within the second view ; we use the differential tracker KLT [21]. Thanks to the stereo rig calibration, a sparse $3D$ model of the object can then be built.

2) Vision-based arm positioning: During the motion, the points are tracked in each optical flow with KLT. The pose estimation is done with a stereo implementation of the virtual visual servoing, as in [5].

The grasping strategy consists of controlling the translational velocities of the arm to move toward the object while centering the box's centroid. Its desired position is about 200 mm from the gripper frame, *i.e.* about 5 cm from the gripper's fingers.

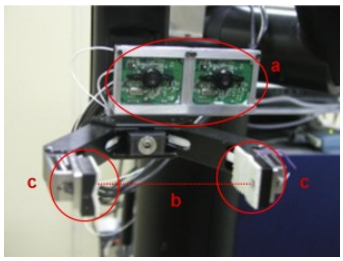

Fig. 7. Stereo rig used to bring the gripper just in front of the object. When the cameras are too close to the object, a blind forward motion is performed so that the object enters the gripper. This is detected by an optical barrier (b). The gripper is then closed, applying a pressure controlled by load cells (c).

Fig. 8. Cup tracking. Only the right image is shown. The first view is the initial one where the box has been defined.

Fig. 9. Example of objects correctly grasped (cards, can, book, bottle).

3) Experiments: Figure 8 illustrates the tracker behavior on a classical object. The box defined by the user is correctly tracked even when the object undergoes rotations.

This technique has been integrated into SAM, and intensively tested during clinical evaluations and several demonstrations. Figure 9 shows a variety of textured objects that have been correctly tracked and grasped. Figure 10 illustrates the position-based control of the arm. It shows the classical exponential decrease of the error.

This method presents two main advantages: (i) it is very easy to launch: only two user clicks are needed to define the box (ii) no $3D$ *a priori* information is required, since all the needed data is automatically extracted from the visual input. Furthermore, the initialization step is not time consuming: once the user has defined the box, the sparse model is estimated in around 100 ms, and the arm guidance toward the object starts almost directly.

However, this grasping strategy fails when the grasping position should be associated to the object's shape and pose, e.g. an object lying on a table or with special features (tea cup with an handle).

In order to obtain a more suited strategy, it is then necessary to extract more information on the object.

B. Rough 3D shape estimation by active vision

The definition of a better grasping position implies to estimate the object shape on-line. We suggest that the objective here is not to get an accurate object reconstruction, but rather

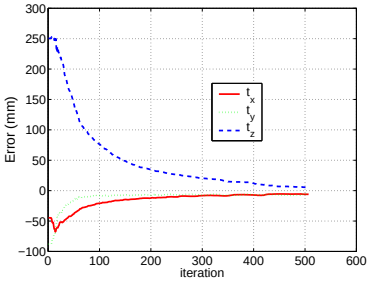

Fig. 10. Visual servoing on the card box (see Fig. 9): object center position error (in mm) vs iteration.

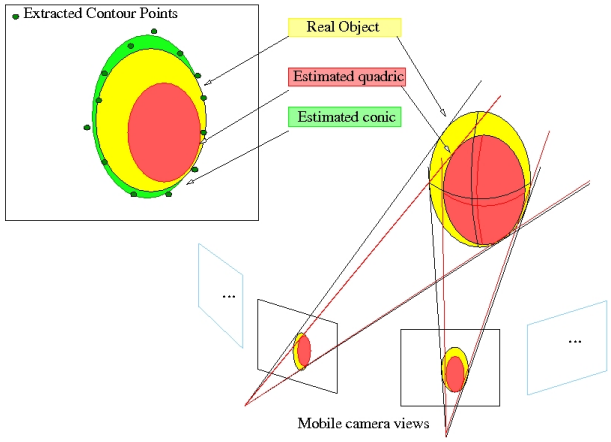

Fig. 11. Quadric fitting scheme. Within each view, the real object shape projection (in yellow) is approximated by a conic (in green). The projection of the estimated quadric is in red. The optimization process consists in reducing the difference between the quadric projections and the measured conics.

to gather enough information, *i.e.* the pose and the rough size of the object, to allow a manipulator to grasp it by aligning the gripper with its minor axis while being perpendicular to its major axis.

This approach is based on contour analysis and on implicit $3D$ reconstruction methods [6]. $3D$ shapes are represented by quadrics. They have the nice property of projecting on an image plane in conics, which provide compact representations that are easy to extract. The reconstruction scheme is the following: get several views of the object at different camera locations, track the conics in the acquired views, and use the parameters of the conics to estimate by minimization the parameters of the corresponding quadric (see Fig. 11). The quality of the reconstruction obviously relies on the locations of each acquired views. Hence we also propose to use active vision in order to determine the next best view.

1) Contour extraction: Active contours are used to extract the points of the object's edge [13]. We use a parametric formulation of the active contour [2] which is more robust than the classical formulation based directly on point motion. The use of such techniques adds two assumptions: (i) the object is entirely seen in every view (it is ensured by the active vision step, see III-B.4), (ii) the object can be segmented from the scene without resorting to either prior knowledge about its appearance or to an *a priori* known model.

As an input, the active contour algorithm needs an initial box almost surrounding the object. This information can be provided by the method used to get the object inside the embedded camera FOV (see previous section). Note that one click is even sufficient. Indeed, if the click is almost at the center of the object, the scale of the box can be automatically obtained by studying the object intrinsic scale [16].

In each view, the active contours extraction gives a set of $2D$ image points $\mathbf{x} = (x, y, 1)$ (in green in fig. 11) that belong to the apparent contour of the object.

2) Conic parameters estimation: The points extracted by the active edge detector are then used to estimate the corresponding $\mathbf{C}_{3\times3}$ conic parameters such that [26]:

$$\mathbf{g}(\mathbf{x}, \mathbf{c}) = \mathbf{x}^\top \mathbf{C} \mathbf{x}, \tag{6}$$

This computation is performed for each considered view, and the obtained $\mathbf{C}_j$ conic parameters are stored along with the corresponding camera positions.

3) Quadric representation: This step consists of estimating the quadric parameters whose projection best fits the data stored in the previous step.

The equation of a quadric expressed in the Cartesian reference frame, $\mathcal{R}_w$, is such that:

$$\mathbf{h}_w({}^w\mathbf{X}, {}^w\mathbf{\Gamma}) = {}^w\mathbf{X}^\top {}^w\mathbf{\Gamma} {}^w\mathbf{X}, \tag{7}$$

where ${}^w\mathbf{X} = (X_w, Y_w, Z_w, 1)$ are the homogeneous $3D$ coordinates of a contour point expressed in $\mathcal{R}_w$, and ${}^w\mathbf{\Gamma}$ is the symmetric positive matrix associated with the quadric.

Given an estimation of the quadric parameters ${}^w\mathbf{\Gamma}$ and the camera calibration (extrinsic and intrinsic parameters), we can compute the corresponding projections $\hat{\mathbf{C}}$ in every view taken by the eye-in-hand camera. Thus, the quadric parametrization that best fits the observed object shape is the one that minimizes the error between the measured conics $\mathbf{C}$ and the projected ones, $\hat{\mathbf{C}}$. This quadric is obtained by minimizing the following cost function:

$$f\left({}^w\mathbf{\Gamma}\right) = \sum_{ij} \frac{(\hat{\mathbf{c}}_{ij} - \mathbf{c}_{ij})^2}{\sigma_{ij}}, \tag{8}$$

where $i \in [0, 5]$ is the index of the i^{th} conic parameter and $j \in [0, N]$ the index of the j^{th} view.

As in [25], we can solve this problem using non-linear minimization techniques. In order to cope with potential noise in the edge points extraction, we propose to us a robust Levenberg-Marquartd minimization algorithm [23].

4) Active vision to cope with ambiguities: The quality of the estimation of the quadric parameters strongly depends on the different views used to describe the object. For instance, views taken too closely to eachother will provide a bad estimation of the quadric.

Active vision is used to define the best camera position to describe the object. [25] proposed to use the uncertainty of the parameter estimation to control the camera displacement. They highlight the link between the uncertainty and the

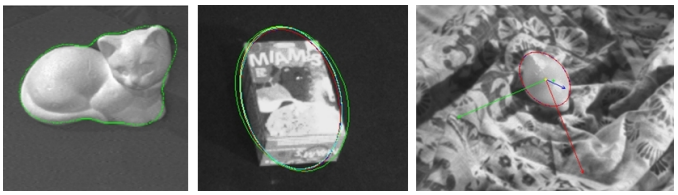

Fig. 12. Object reconstruction results: the two first images are examples of active contours. Last image illustrates the final object frame estimation. The blue and red arrays are respectively the major and minor axis of the object.

covariance matrix on the quadric parameters resulting from the optimization process. The basic idea is to move the camera to the position that generates the most information about the most poorly estimated parameters.

Instead of computing the minimum of the determinant of the covariance matrix like in [25], we select the next best view by minimizing the Froebenius norm of the covariance matrix. It is indeed less time consuming and has the advantage of avoiding the local minima that occur as soon as one of the parameters is well estimated.

Here, we face a non linear minimization problem without analytic Jacobian computation. Thus the optimisation is done with the Simplex method of Nelder and Mead [23].

This method is used to compute the translational components of the camera velocity. The rotational component is deduced by visual servoing [9], so that the projection of the centroid of the estimated quadric remains in the center of the image plane.

5) Experimental results: A frame attached to the object can be computed directly from the parameters of the estimated quadric, as shown in Figure 12.

At the end of the reconstruction process, the gripper is aligned with the object frame using $3D$ servoing and then moved toward the object in order to grasp it. The quadric parameters can be continuously refined until gripper closure. Since our reconstruction process is directly based on object contour extraction in the images, the solution is very fast, allowing us to compute the object shape in real time and to use it in a closed-loop grasping task. The proposed solution is fully generic and works for any roughly convex object. We are currently integrating this grasping procedure on the SAM platform (current experiments use the Afma6 arm).

IV. Conclusion

This paper has presented different solutions to orientate a robotic arm in the direction of an object and then to grasp it. In all the techniques proposed, we have minimized the assumptions on the grasping environment and on the object appearance, so that the system can handle a wide range of situations. The use of our solutions does not require the user to have any technical expertise, and needs a very small number of clicks. Furthermore, the solutions for the two problems addressed can easily be combined, depending on the robot structure, the user need, and/or convenience.

Some of these techniques have been evaluated by disabled subjects with a static robot. We are preparing evaluations involving a mobile unit. The methods validated on robotic cells are currently integrated on SAM, and will be soon tested by the envisioned end-users.

References

[1] R. Alqasemi, E. Mccaffrey, K. Edwards, and R. Dubey. Wheelchair-mounted robotic arms: analysis, evaluation and development. In IEEE ICAIM, pages 1164–1169, Monterey, USA, July 2005.

[2] P. Brigger, J. Hoeg, and M Unser. B-spline snakes: a flexible tool for parametric contour detection. IEEE *Trans. on Image Processing*, 9:1484–1496, Sep. 2000.

[3] F. Chaumette and S. Hutchinson. Visual servo control, part I: Basic approaches. IEEE RAM, 13(4):82–90, Dec. 2006.

[4] F. de Chaumont, B. Marhic, and L. Delahoche. Omnidirectional mirror design: multiple linear ring-windows viewing. In IEEE *Inter. Symp. on Industrial Electronics*, pages 311–316, Ajaccio, France, May 2004.

[5] F. Dionnet and E. Marchand. Robust stereo tracking for space robotic applications. In IEEE IROS, pages 3373–3378, San Diego, USA, Oct. 2007.

[6] C. Dune, E. Marchand, C. Collewet, and C. Leroux. Active rough shape estimation of unknown objects. In IEEE IROS, Nice, France, Sep. 2008.

[7] C. Dune, E. Marchand, and C. Leroux. One click focus with eye-in-hand/eye-to-hand cooperation. In IEEE ICRA, pages 2471–2476, Roma, Italy, Apr. 2007.

[8] H. Eftring and K. Boschian. Technical results from manus user trials. In ICORR, pages 136–141, Standford, USA, July 1999.

[9] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. IEEE *Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[10] G. Flandin and F. Chaumette. Visual data fusion for objects localization by active vision. In ECCV, pages 312–326, Copenhagen, Denmark, May 2002.

[11] A. Hauck, J. Ruttinger, M. Sorg, and G. Farber. Visual determination of 3d grasping points on unknown objects with a binocular camera system. In IEEE IROS, pages 272–278, Kyongju, South Korea, Oct. 1999.

[12] R. Horaud, D. Knossow, and M. Michaelis. Camera cooperation for achieving visual attention. *Machine Vision and Applications*, 16(6):331–342, Feb. 2006.

[13] M. Kass, Witkin A., and D. Terzopoulos. Snakes: Active contour models. IJCV, 1:321–333, Sep. 1987.

[14] D. Kragic, M. Bjorkman, H. Christensen, and J.-0. Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and Autonomous Systems*, 52(1):85–100, July 2005.

[15] F. Liefhebber and J. Sijs. Vision-based control of the manus using SIFT. In ICORR, pages 854–861, Noordwijk, The Netherlands, June 2007.

[16] T. Lindeberg. Feature detection with automatic scale selection. IJCV, 26(3):171–189, Nov. 1998.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91–110, Nov. 2004.

[18] A. Remazeilles, C. Leroux, and G. Chalubert. SAM: a robotic butler for handicapped people. In IEEE ROMAN, Munich, Germany, Aug. 2008.

[19] P.J. Sanz, A. Requena, J.M. Iesta, and A.P. del Pobil. Grasping the not-so-obvious: vision-based object handling for industrial applications. IEEE RAM, 12(3):44–52, Sep. 2005.

[20] A. Saxena, J. Driemeyer, and Y. Ng. Robotic grasping of novel objects using vision. IJRR, 27:157–173, Feb. 2008.

[21] J. Shi and C. Tomasi. Good features to track. In IEEE CVPR, pages 593–600, Seattle, USA, June 1994.

[22] D. Taddeucci and P. Dario. Experiments in synthetic psychology for tactile perception in robots: steps towards implementing humanoid robots. In IEEE ICRA, volume 3, pages 2262–2267, Leuven, Belgium, May 1998.

[23] W. Vetterling, S. Teukolsky, W. Press, and B. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, Feb. 2002.

[24] I. Volosyak, O. Ivlev, and A. Graser. Rehabilitation robot FRIEND II - the general concept and current implementation. In ICORR, pages 540–544, Chicago, USA, June 2005.

[25] P. Whaite and Ferrie F.P. Autonomous exploration driven by uncertainty. IEEE PAMI, 19(3):193 – 205, Mar. 1997.

[26] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report RR-2676, INRIA, Oct. 1997.